



J. Frank Osha  
T 202.663.7915  
fosha@sughrue.com

February 19, 2002

BOX PATENT APPLICATION  
Commissioner for Patents  
Washington, D.C. 20231

#3  
2100 Pennsylvania Avenue, NW  
Washington, DC 20037-3213  
T 202.293.7060  
F 202.293.7860

1010 El Camino Real  
Menlo Park, CA 94025-4345  
T 650.325.5800  
F 650.325.6606

Toei Nishi Shimbashi Bldg. 4F  
13-5 Nishi Shimbashi 1-Chome  
Minato-Ku, Tokyo 105-0003  
Japan  
T 03.3503.3760  
F 03.3503.3756

www.sughrue.com

Re: Application of Ryuji SATO  
METHOD THAT CAUSES PROGRAM ANALYSIS OF DEVICE DRIVER TO  
BECOME DIFFICULT  
Assignee: NEC CORPORATION  
Our Ref. Q68583

10/076404  
PTO  
02/19/02

Dear Sir:

Attached hereto is the application identified above comprising 17 sheets of the specification, including the claims and abstract, 6 sheets of drawings, executed Assignment and PTO 1595 form, and executed Declaration and Power of Attorney. Also enclosed is an Information Disclosure Statement.

The Government filing fee is calculated as follows:

Total claims	11 - 20	=		x	\$18.00	=	\$0.00
Independent claims	3 - 3	=		x	\$84.00	=	\$0.00
Base Fee							\$740.00

<b>TOTAL FILING FEE</b>	<b>\$740.00</b>
Recordation of Assignment	\$40.00
<b>TOTAL FEE</b>	<b>\$780.00</b>

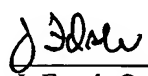
A check for the statutory filing fee of \$740.00 is attached. You are also directed and authorized to charge or credit any difference or overpayment to Deposit Account No. 19-4880. The Commissioner is hereby authorized to charge any fees under 37 C.F.R. §§ 1.16 and 1.17 and any petitions for extension of time under 37 C.F.R. § 1.136 which may be required during the entire pendency of the application to Deposit Account No. 19-4880. A duplicate copy of this transmittal letter is attached.

Priority is claimed from:

<u>Country</u>	<u>Application No</u>	<u>Filing Date</u>
Japan	2001-043748	February 20, 2001

The priority document is enclosed herewith.

Respectfully submitted,  
SUGHRUE MION, PLLC  
Attorneys for Applicant

By:   
J. Frank Osha  
Registration No. 24,625

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

JCS711 U.S. PTO  
10/076404  
02/19/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office

出 願 年 月 日  
Date of Application:

2001年 2月20日

出 願 番 号  
Application Number:

特願2001-043748

出 願 人  
Applicant(s):

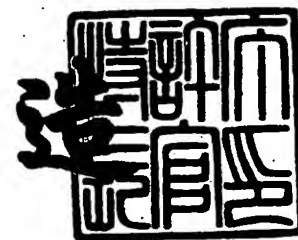
日本電気株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年11月26日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3102923

【書類名】 特許願

【整理番号】 68501909

【提出日】 平成13年 2月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/06

【発明者】

    【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内

    【氏名】 佐藤 隆士

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100065385

    【弁理士】

    【氏名又は名称】 山下 穰平

    【電話番号】 03-3431-1831

【手数料の表示】

    【予納台帳番号】 010700

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9001713

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 デバイスドライバ作動方法

【特許請求の範囲】

【請求項 1】 デバイスドライバの主処理のプログラムコード部分を予め暗号化し、

前記デバイスドライバの初期化处理において前記プログラムコード部分を復号化し、

前記プログラムコード部分を実行した後、前記デバイスドライバ解放時に、前記プログラムコード部分を再び暗号化することを特徴とするデバイスドライバ作動方法。

【請求項 2】 デバイスドライバの主処理のプログラムコード部分を予め暗号化し、

前記デバイスドライバを初期化し、

初期化处理終了後に、前記プログラムコード部分を復号化し、

前記プログラムコード部分を実行終了後に、前記プログラムコード部分を再び暗号化し、

その後に、前記デバイスドライバを解放することを特徴とするデバイスドライバ作動方法。

【請求項 3】 デバイスドライバの主処理のプログラムコード部分を第 1 暗号化鍵で予め暗号化した後、更に第 2 暗号化鍵で暗号化し、

前記デバイスドライバの初期化处理において第 1 暗号化鍵で暗号化された前記プログラムコード部分を第 1 復号化鍵で復号化し、

初期化处理終了後に、第 2 暗号化鍵で暗号化された前記プログラムコード部分を第 2 復号化キーで復号化し、

前記プログラムコード部分を実行終了後に、前記プログラムコード部分を再び第 2 暗号化キーで暗号化し、

前記プログラムコード部分を実行した後、前記デバイスドライバ解放時に、前記プログラムコード部分を再び第 1 暗号化鍵で暗号化することを特徴とするデバイスドライバ作動方法。

【請求項 4】 アプリケーション側に 1 又は 2 以上のメモリ領域を設け、  
前記メモリ領域の一つに格納された数値に基づいて、前記暗号化及び前記復号化のための鍵を作成することを特徴とする請求項 1 乃至 3 記載のデバイスドライバ作動方法。

【請求項 5】 アプリケーションとデバイスドライバの間で認証を行うことを特徴とする請求項 1 乃至 3 記載のデバイスドライバ作動方法。

【請求項 6】 アプリケーションは、デバイスドライバに出力データを渡す前に、デバイスドライバの前記プログラムコード部分が改竄されているか否かを検出し、改竄されている場合には、ハードウェアへの前記出力データの出力を停止し、

前記デバイスドライバは、前記アプリケーションに入力データを渡す前に、アプリケーションの前記プログラムコード部分が改竄されているか否かを検出し、改竄されている場合には、アプリケーションへの前記入力データの出力を停止することを特徴とする請求項 1 乃至 3 記載のデバイスドライバ作動方法。

【請求項 7】 前記デバイスドライバは、アプリケーションの暗号化データを復号化せず、

前記アプリケーションは、前記改竄がない場合にのみ、前記暗号化データを復号して前記デバイスドライバに出力することを特徴とする請求項 6 記載のデバイスドライバ作動方法。

#### 【発明の詳細な説明】

【0001】

#### 【発明が属する技術分野】

本発明は、第三者による解析を妨げるデバイスドライバの構築方法に関する。

【0002】

#### 【従来の技術】

従来、データやソフトウェアの改竄や不正利用を防止する技術が考えられている。たとえば、特開 2 0 0 0 - 1 2 2 8 6 1 号公報の「データ等の不正改竄防止システム及びそれと併用される暗号化装置」においては、暗号化されたプログラムコードを複数のブロックに分割する。そして、次に実行するブロックを暗号解

除するためにハッシュ関数等を用いて暗号鍵を計算する。従って、不正利用者がプログラムコードの一部を改竄すれば正しい暗号鍵が得られず、プログラムの実行が呈するようになっている。

## 【0003】

又、特開2000-347848号公報の「半導体IC、情報処理方法、情報処理装置、並びにプログラム格納体」においては、コンテンツを暗号化して記録するとともに、暗号鍵も保存用鍵で暗号化した上で記録する。従って、コンテンツをコピーしても、これを復号することはできないので、コピーが大量に配布されることを防止することができる。又、パーソナルコンピュータが外部機器にデータを渡すとき、その前に相互認証を行うため、不正な外部機器にデータを渡してしまうようなことが防止される。又、外部機器からパーソナルコンピュータにデータを渡すとき、その前にパーソナルコンピュータのソフトウェアが正当なものであるかを相互認証により確認するため、不正なコンピュータにデータを渡してしまうようなことが防止される。なお、相互認証は認証局を介して行なわれる。

## 【0004】

又、特開2000-347852号公報の「情報処理装置及び方法、並びにプログラム格納体」においては、パーソナルコンピュータにおけるソフトウェア機能のうち、所定の部分を外付けのアダプタに負担させている。従って、パーソナルコンピュータのソフトウェアを解析しただけでは、全体としてどのような処理となっているか分からない。従って、不正利用者が、自らの意図に合わせてソフトウェアを改竄することが困難となる。又、パーソナルコンピュータのソフトウェアプログラムをプログラム暗号化鍵で暗号化し、プログラムの実行に必要なデータをアダプタが生成するデータ暗号化鍵で暗号化する。従って、プログラムのみをCD-ROM等で受け取っても、他のアダプタで実行することはできない。

## 【0005】

又、特開平3-276345号公報の「マイクロコントローラ」においては、スクランブルされたプログラム又はデータをメモリに格納するとともに、これを解読するキーデータを別のメモリに格納している。

## 【0006】

又、特開平11-39158号公報の「実行プログラムの保護法方およびその装置」は、1チップの中にFROM（フラッシュメモリ）等のROMとRAMと処理部が形成されているLSIに関する。従来、このようなLSIでは、RAMの内容をコピーすれば、処理装置の秘密鍵の読み出しを阻止できないという欠点がある。そこで、実行プログラムの改竄を検出し、改竄が検出されれば直ちに実行を停止するようにしている。具体的には、実行処理プログラムを編集したときの編集時メッセージダイジェストと、暗号処理された実行処理プログラムの中の第2のメッセージダイジェストとを照合し、一致しなかった場合には、実行プログラムが改竄されたと判断する。ここに、メッセージダイジェストは、一方向性ハッシュ関数によって実行プログラムを処理した後における最後の16バイトのデータである。

【0007】

【発明が解決しようとする課題】

ところで、デバイスドライバは、ハードウェアとOS(Operating System)の間で、OSの仕様に合わせてデータの入出力を制御する、特殊なプログラムである。デバイスドライバによって異なるメーカー間の仕様の違いを吸収することができ、ハードウェアを同様の手順によって使用することができる。一般的なOSにおいては、ビデオカードやサウンドカードなど、異なるメーカーの同じ種類のハードウェアを、共通化して扱えるようにするために、デバイスドライバを仲介させる方式をとっている。メーカーにデバイスドライバを用意させて、OS上のアプリケーションからハードウェアをアクセスする方法を統一しておく。このときデバイスドライバによってデバイスへの入出力方法の違いが吸収される。この仕組みにより、例えばアプリケーション側でビデオカードの種類ごとに違うプログラムを用意しなくてすむようになる。デバイスドライバは直接ハードウェアを扱う特殊なプログラムなので、アプリケーションと違って、メモリアクセスや入出力命令の制限のない特権レベルで実行されている。

【0008】

近年、映像データや音声データなどは、コピー防止や著作権保護といった観点から、データに暗号化を施して取り扱うことが多くなってきた。暗号化されたデータは再生用アプリケーションが元のデータに復号してから、デバイスドライバ

に出力する。すなわち、特にビデオカードやサウンドカード用のデバイスドライバは、音楽や音声データに施されていた暗号化が解除された状態のデータを受け取るので、悪意ある第三者がデバイスドライバを改ざんしたり、データの入力部分にトラップを仕掛けることで、復号されたデータを外部記憶装置に記録したり、復号データと暗号化データを比較することで暗号化アルゴリズムが解明されたり、鍵を知られるといったことが起こりうる。

#### 【0009】

第三者が、このようなことを行うには、そのハードウェアへのアクセス手順などを知っている必要がある。デバイスドライバは、手続きに従って、データをハードウェアに通知するという性質から、アプリケーションのように装飾部分が必要なく、これらのプログラムよりもシンプルな構造になっている。

#### 【0010】

そのため、第三者がデバイスドライバをリバースエンジニアリングすることで、アクセス手順などハードウェアなどの情報を知ることがたやすいという状況になっている。

#### 【0011】

そこで、本発明は、デバイスドライバのプログラム解析を困難にするデバイスドライバ構築方法を提供することを課題としている。

#### 【0012】

##### 【課題を解決するための手段】

上記課題を解決するための本発明のデバイスドライバの作動方法においては、デバイスドライバの主処理のプログラムコード部分を予め暗号化し、上記デバイスドライバの初期化处理において上記プログラムコード部分を復号化し、上記プログラムコード部分を実行した後、上記デバイスドライバ解放時に、上記プログラムコード部分を再び暗号化するようにしている。

#### 【0013】

又、本発明においては、デバイスドライバの主処理のプログラムコード部分を予め暗号化し、上記デバイスドライバを初期化し、初期化处理終了後に、上記プログラムコード部分を復号化し、上記プログラムコード部分を実行終了後に、上



記プログラムコード部分を再び暗号化し、その後に、上記デバイスドライバを解放してもよい。

【0014】

又、本発明においては、デバイスドライバの主処理のプログラムコード部分を第1暗号化鍵で予め暗号化した後、更に第2暗号化鍵で暗号化し、上記デバイスドライバの初期化処理において第1暗号化鍵で暗号化された上記プログラムコード部分を第1復号化鍵で復号化し、初期化処理終了後に、第2暗号化鍵で暗号化された上記プログラムコード部分を第2復号化キーで復号化し、上記プログラムコード部分を実行終了後に、上記プログラムコード部分を再び第2暗号化キーで暗号化し、上記プログラムコード部分を実行した後、上記デバイスドライバ解放時に、上記プログラムコード部分を再び第1暗号化鍵で暗号化するようにしてもよい。

【0015】

すなわち、本発明においては、デバイスドライバで解析から保護したい手続き部分、主処理のプログラムコード部分にあらかじめ暗号化をかけておき、初期化ルーチンでこの暗号化を復号する。ドライバ解放時にはふたたび暗号化する。この復号化と再暗号化の際に使われる鍵は、デバイスドライバとアプリケーションの間で共通の領域を用いて、そこで数回の演算で得たものを用いる。これはデバイスドライバが、メモリアクセスに制限のない特権モードで動作することを利用している。

【0016】

【発明の実施の形態】

以下、図面を参照して本発明の実施の形態について説明する。

【0017】

〔実施形態1〕

図1は、一般的なOSにおけるデバイスドライバの役割について示した図である。ユーザレベルのアプリケーション10から、デバイスドライバ11にデータが渡される。デバイスドライバ11は、これをメーカ固有の手続きでハードウェア12に出力する。入力の場合はハードウェア12からのデータをデバイスドライ

バ 1 1 が 0 S の仕様に従って アプリケーション 1 0 に渡す。

デバイスへの出力が行われる場合は、アプリケーション 1 0 から渡された出力データを、デバイスドライバ 1 1 がハードウェア 1 2 の仕様に従って、出力を行う。アプリケーション 1 0 は、メモリアクセスなどに制限のあるユーザレベルプログラムである。デバイスドライバ 1 1 は、ハードウェア 1 2 と直接データのやり取りを行う必要があるので特権レベルが与えられている。特権レベルは、メモリアクセスやハードウェアへの入出力に制限のない権限レベルである。

#### 【 0 0 1 8 】

図 2 は、第三者によってデバイスドライバが改ざんされる場合について説明した図である。たとえば、著作権保護されているコンテンツ 2 0 をアプリケーション 2 1 によって再生するものとする。この場合、コンテンツの暗号化されたデータはアプリケーション 2 1 によって復号された状態でデバイスドライバを経由してハードウェア 2 3 へと渡される。このときデバイスドライバに渡されるデータは暗号化されていないので、第三者が改ざんされたデバイスドライバ 2 2 を使用することによって、暗号化されていないデータをそのまま外部記憶装置 2 4 へ保存することができる。

#### 【 0 0 1 9 】

図 3 は、一般的なデバイスドライバの構造である、デバイスドライバ 3 0 は初期化处理 3 1、主処理 3 2、終了処理 3 3 を含んでいる。初期化处理はデバイスドライバ起動時に行われる処理、主処理はアプリケーションとデバイス間のデータの入出力処理、終了処理はデバイスドライバ解放前に行われる処理を行う。デバイスドライバはデバイスが接続されたときに初期化处理 3 1 を実行する。そのあとデバイス使用時には主処理がデータの仲介を行い、終了時は終了処理 3 3 においてデバイスの解放処理を行う。

#### 【 0 0 2 0 】

図 4 は、本発明によって保護されたデバイスドライバである。図 3 で主処理にあたるプログラムコード部分が暗号化されているので、第三者はデバイスドライバを逆アセンブラで解析することはできない。初期化处理 4 1 において、この暗号化コード 4 2 は復号化される。このためデバイス使用時には、暗号化コード 4

2は元のプログラムコードに戻っているので、通常の処理を行うことができる。デバイス解放時には終了処理43において、プログラムコードから暗号化コード42に戻される。

図5 (A) は、初期化処理を説明するためのフローチャートである。デバイスドライバはイベント駆動型のものを想定している。

【0021】

アプリケーションはデバイスドライバにデータの入出力を依頼し、デバイスドライバの起動の依頼をOSから受けて、デバイスドライバのプログラムがロードされ、初期化のイベントが発行される。これにより初期化処理が行われる（ステップA1）。

【0022】

次に、アプリケーションのメモリ上のある位置から数値を取り出して演算を行い、暗号解除に用いる鍵にする（ステップA2）。

【0023】

次に、暗号化コード42を復号化して、実行可能なプログラムコードの状態に戻して、初期化処理を終了する（ステップA3）。

【0024】

図5 (B) は、データアクセス処理を説明するためのフローチャートである。アプリケーションからハードウェアにデータを渡すように依頼されたときに、データアクセス通知のイベントが発行される。すでに暗号解除された暗号化コード42は実処理部のコードになっているので、これを実行する（ステップA4）。

【0025】

デバイスドライバ終了の依頼をOSから受けて、終了処理のイベントが発行される。

【0026】

図5 (C) は、終了処理を説明するためのフローチャートである。

【0027】

デバイスドライバは、アプリケーションのメモリ上のある位置から数値を取り出して演算を行い、暗号の復号に用いる鍵を得る（ステップA5）。

## 【0028】

次に、復号された状態の実処理部のプログラムコードを再び暗号化する（ステップA6）。

## 【0029】

次に、デバイスドライバ終了に必要な終了処理を行う（ステップA7）。

## 【0030】

終了処理のときに実処理部を再び暗号化する必要があるのは、多くの場合デバイスドライバが解放されても、メモリ上にはデバイスドライバのプログラムコードがそのまま残っているからである。この残ったコードを解析されることのないように、終了時に再び暗号化を行う。

## 【0031】

ここで、ステップA2やA5で行う、アプリケーション上のメモリから鍵を得る方法を説明する。

## 【0032】

初期化処理の際にアプリケーションは、デバイスドライバに鍵の作成に使用する領域を通知する。デバイスドライバは特権モードが与えられているので、アプリケーション側の任意のメモリ領域のアクセスが可能である。この鍵の作成領域には初期値が設定されている。これにあらかじめ決めておいた演算を数回行うことで得た数値を鍵とする。

## 【0033】

## 〔実施形態2〕

暗号化の解除を主処理を実行する直前に行い、再暗号化を主処理を実行した直後にしてもよい。これにより、実行速度が低下する代わりに、暗号化が解除されている時間が非常に短くなるため、解析されにくいドライバを構築することができる。

## 【0034】

図6において、デバイスドライバ60は、初期化処理61、主処理62、終了処理64を含む。主処理の実際に保護すべき処理部分だけを暗号化コード63として暗号化する。主処理が行われると、まず暗号化コード63が復号され、実行

される。これが終了するとただちに、再暗号化が行われる。第一の実施形態と異なり、復号を初期化処理 6 1 で行わず、再暗号化を終了処理 6 4 で行わない。主処理の中で復号と暗号化を完結されることによって、主処理の実行のたびに復号と暗号化が行われるようになり、処理速度は低下する。そのかわりに、保護すべき処理部分は一瞬しか暗号化を解除されていないので、解析が非常に困難になる。

## 【 0 0 3 5 】

## 〔実施形態 3〕

初期化時に暗号化コードの復号化と終了時に再暗号化を行い、主処理部を実行する直前に二段目の暗号解除を行い主処理終了後に再暗号化を行うようにしてもよい。これにより、暗号化の手段が二段になるので強固なデバイスドライバを構築することができる。このとき、二段目の暗号化は展開速度を重視して強度より速度を優先した暗号化・復号化を行い、初期化時の一段目の暗号化は強度を重視した暗号化を行うことで、逆アセンブラなどの解析行為に対して強固なシステムを構築することができる。

## 【 0 0 3 6 】

## 〔実施形態 4〕

アプリケーションとデバイスドライバの間で鍵作成に使用する領域は、アプリケーション側に複数用意してもよい。用意した  $n$  個の領域それぞれに演算を行い、それらのうちのひとつだけを鍵として使用する。どれを鍵とするかは始めから決めておく。残りの領域は第三者を混乱させるためのダミーになる。

## 【 0 0 3 7 】

## 〔実施形態 5〕

アプリケーションとデバイスドライバの間で、認証を行ってもよい。鍵を作成するためにアプリケーション側とデバイスドライバとの間で使用する領域を利用して、認証を行うことができる。これにより、デバイスドライバは特定のアプリケーションとのみやりとりを行っていることを常時確認できる。

## 【 0 0 3 8 】

## 〔実施形態 6〕

アプリケーションとデバイスドライバの間で、改ざん検出を行ってもよい。

【0039】

図7に示すように、アプリケーションとデバイスドライバの間で、お互いに改ざんされていないかを検出する。アプリケーション70はデバイスドライバ71にコンテンツのデータを出力する前に、デバイスドライバ71のプログラムコードが改ざんされていないかを調べる。改ざんされていたら、ハードウェア72へのデータの出力を停止する。デバイスドライバ71はアプリケーション70にデータを渡す前に、アプリケーション70のプログラムが改ざんされていないかを調べる。改ざんされていたら、ハードウェア72からのデータを入力するのを停止する。この改ざんの検査には、プログラムコードの値を入力としたハッシュ値を使うのが適当である。これにより、アプリケーションとデバイスドライバは、お互いが改ざんされていないことを確認してからデータを出力することができる。

【0040】

[実施形態7]

デバイスドライバはすべての暗号解除を行わずに、全部もしくは一部のコンテンツのデータにかけられている暗号をそのままにしておく。暗号化されたコンテンツのデータは、改竄がない場合にのみ、アプリケーションによって暗号化を解除されてから、デバイスドライバにデータを出力する。これにより、ハードウェアへのアクセス方法が解析され、不正なデバイスドライバが作成されても、データの暗号化解除方式が判明しなければ、第三者がデータを取得することは困難になる。

【0041】

【発明の効果】

以上説明した本発明によれば、第三者によってデバイスドライバの解析や改ざんを行わせないようにすることができる。その理由は、データアクセス部分のプログラムを暗号化しておき、デバイスドライバ起動時に復号し、終了時に再暗号化することで、解析をさせない仕組みになっていることにある。このときの復号化と暗号化に使う鍵は、アプリケーション側で用意された領域を使用して、お互

いに演算を行うことで得る。デバイスドライバは特権モードであるのでアプリケーション側の任意の領域を読み書きすることができる。これを行うことで暗号解除の鍵を分散させ、解読を妨げる。

【図面の簡単な説明】

【図 1】

デバイスドライバとハードウェアとアプリケーションの関係を示した一般的な OS の図である。

【図 2】

デバイスドライバの改ざんによってデータがコピーされることを示した図である。

【図 3】

一般的なデバイスドライバの構成を示した図である。

【図 4】

本発明を適用したデバイスドライバの構成を示した図である。

【図 5】

デバイスドライバ実行のフローチャートである。

【図 6】

直前に復号と暗号化を行う場合を示した図である。

【図 7】

アプリケーションとデバイスドライバが互いに改ざん検査を行うことを示した図である。

【符号の説明】

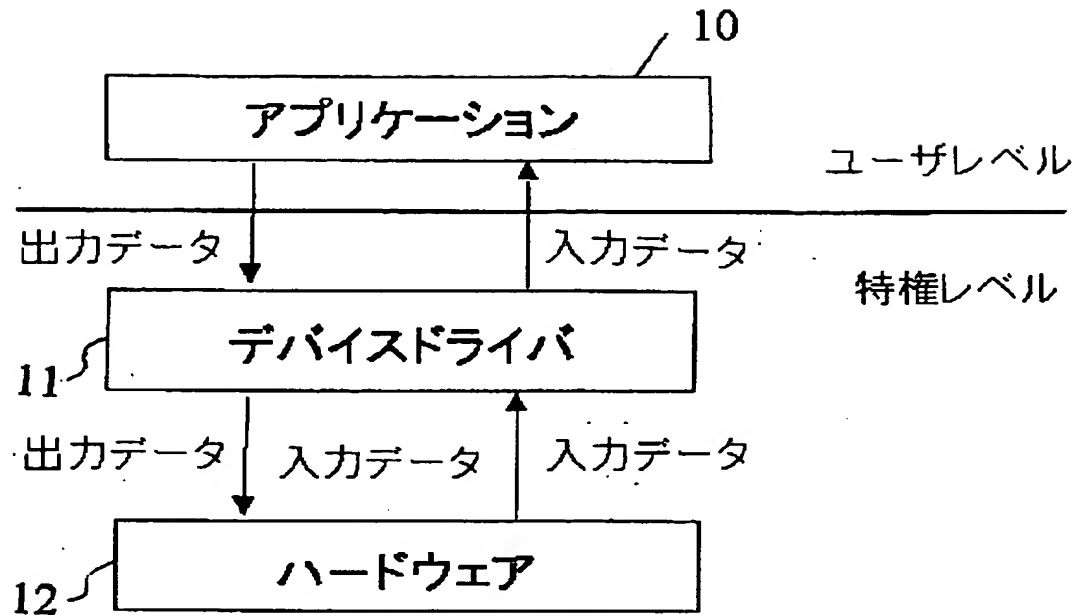
- 1 0 アプリケーション
- 1 1 デバイスドライバ
- 1 2 ハードウェア
- 2 0 コンテンツ
- 2 1 アプリケーション
- 2 2 改ざんされたデバイスドライバ
- 2 3 ハードウェア

- 2 4 外部記憶装置
- 3 0 デバイスドライバ
- 3 1 初期化处理
- 3 2 主処理
- 3 3 終了処理
- 4 0 デバイスドライバ
- 4 1 初期化处理
- 4 2 主処理
- 4 3 終了処理
- 6 0 デバイスドライバ
- 6 1 初期化处理
- 6 2 主処理
- 6 3 暗号化コード
- 6 4 終了処理
- 7 0 アプリケーション
- 7 1 デバイスドライバ
- 7 2 ハードウェア

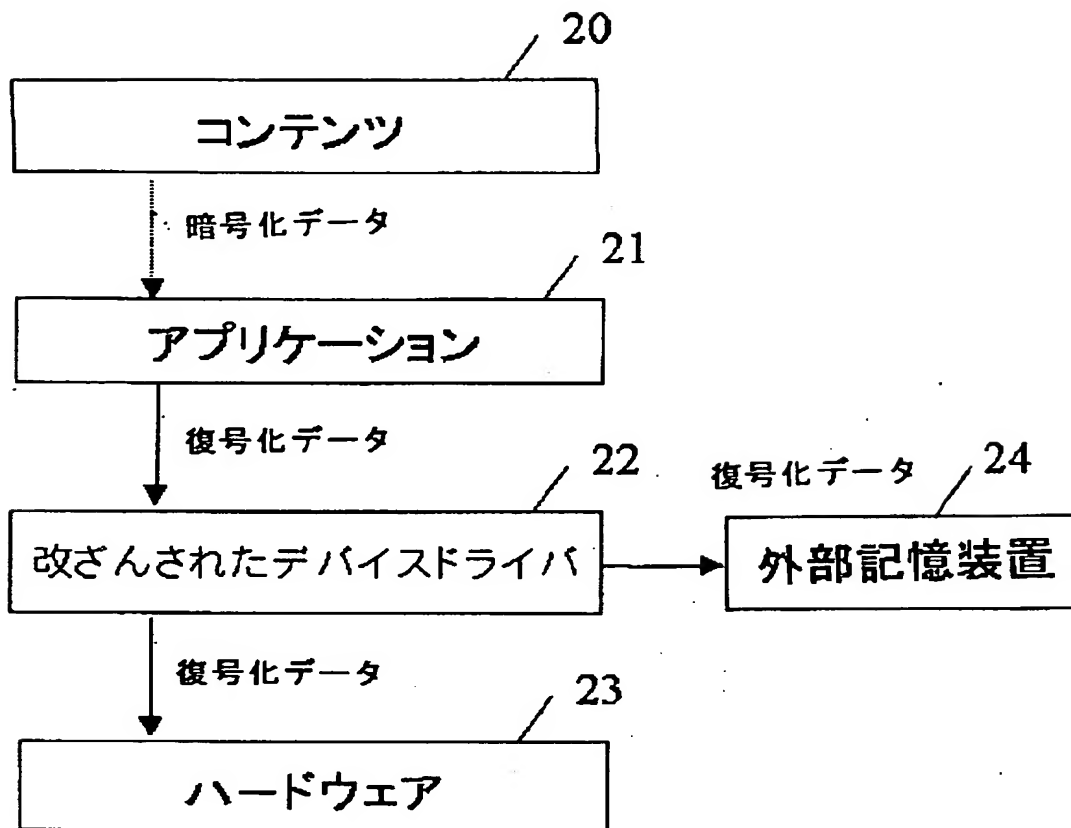


【書類名】 図面

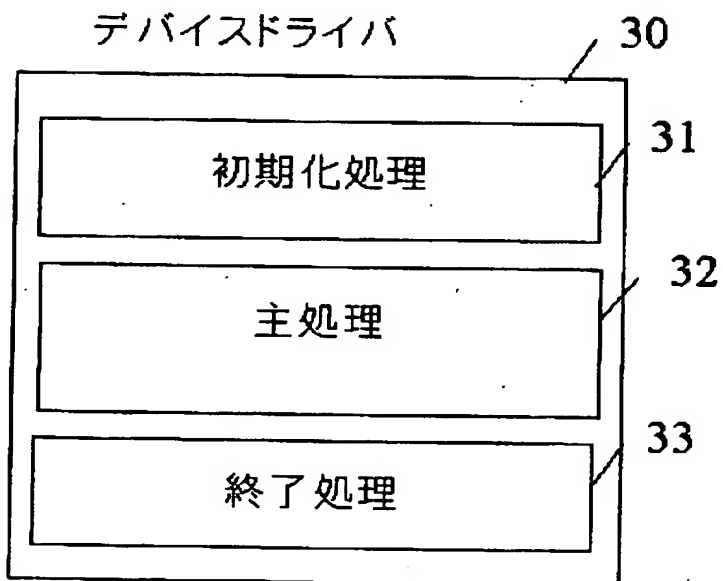
【図1】



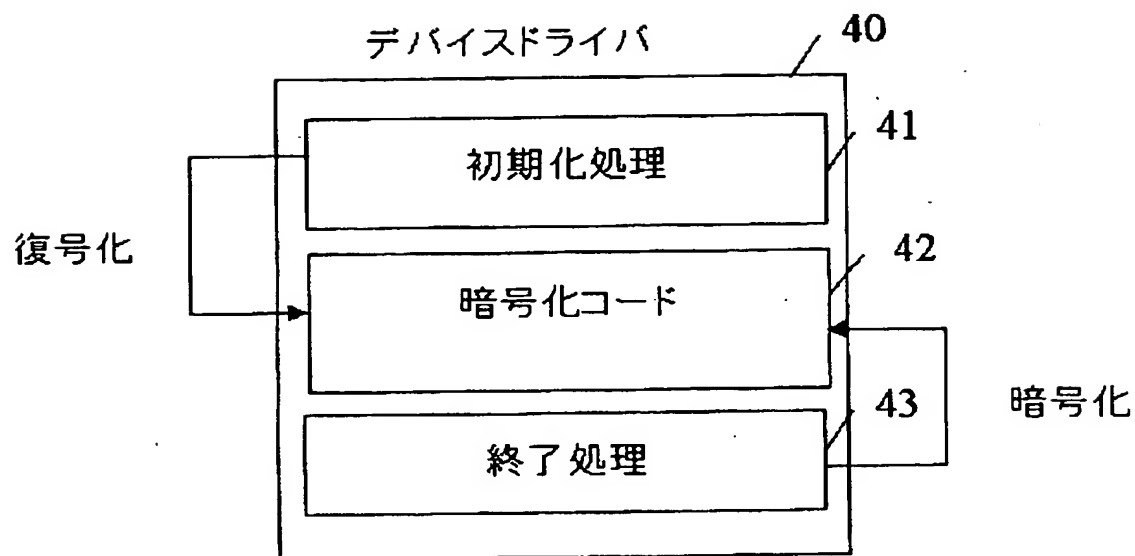
【図 2】



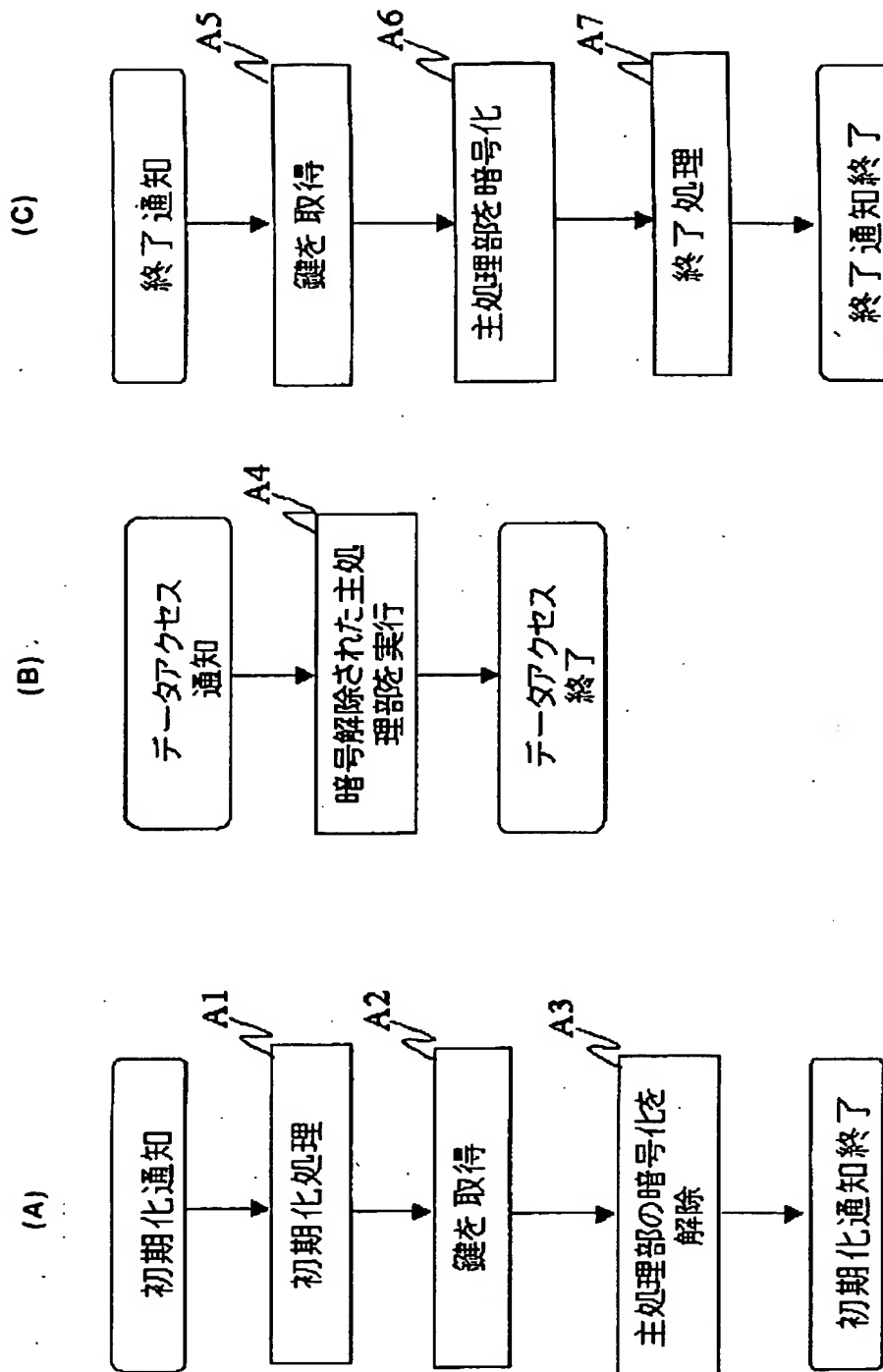
【図 3】



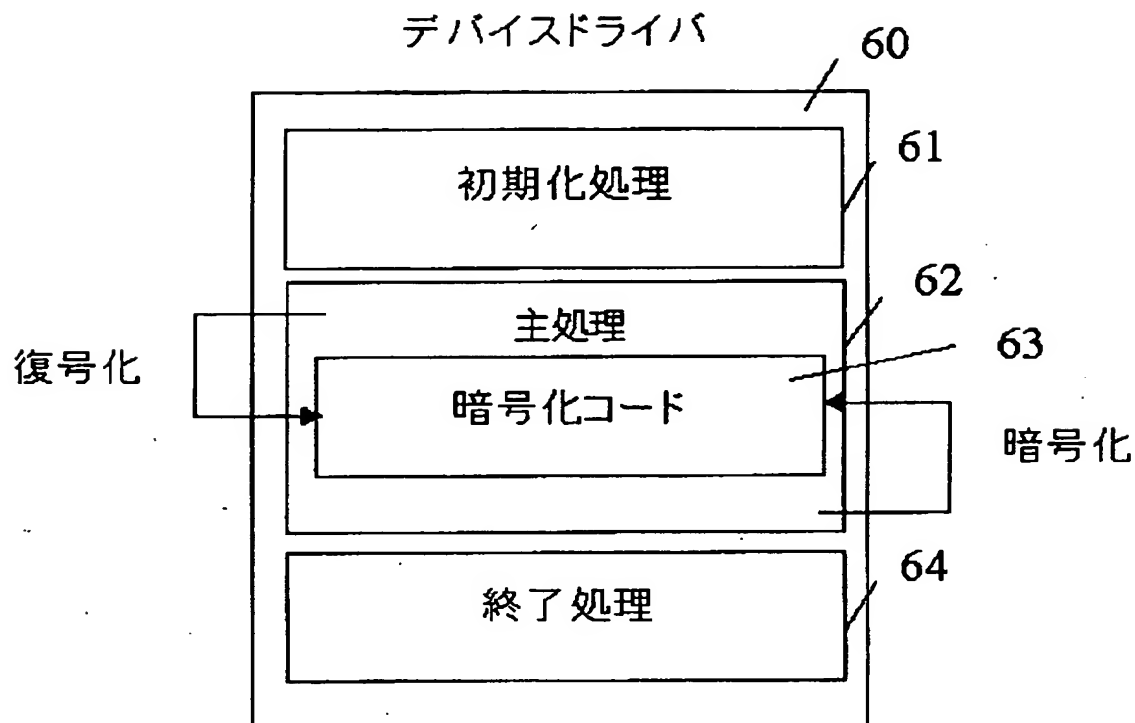
【図 4】



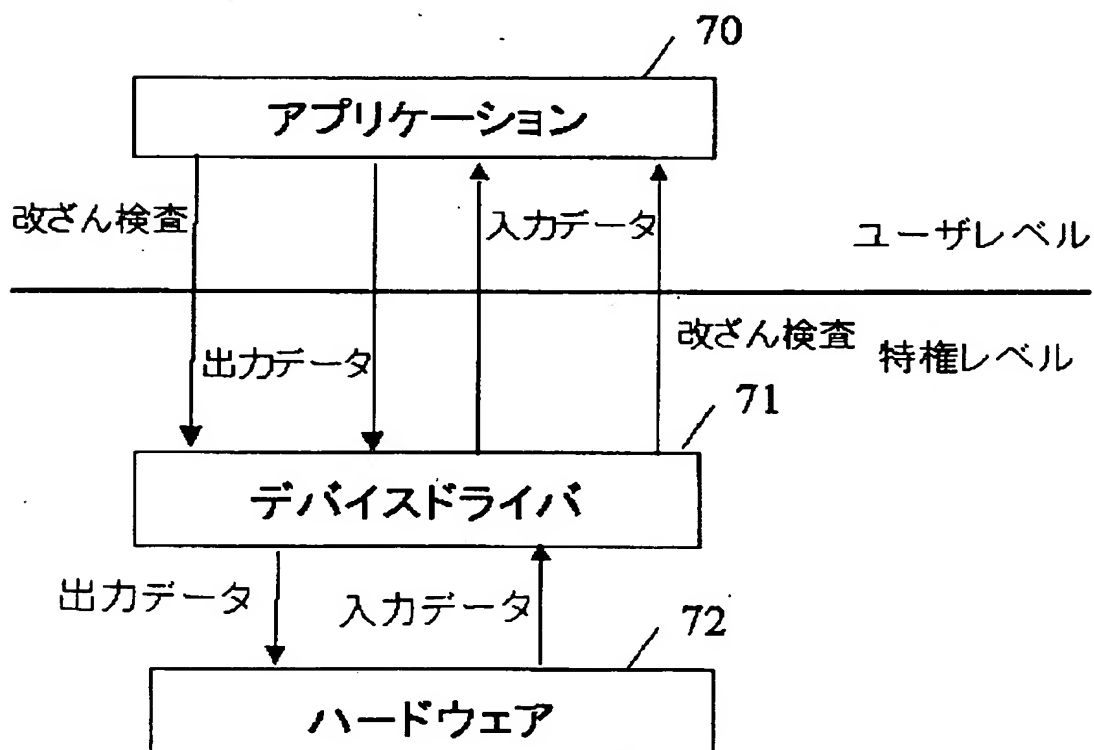
【図5】



【図 6】



【図 7】



【書類名】 要約書

【要約】

【課題】 デバイスドライバのプログラム解析を困難にするデバイスドライバ構築方法を提供する。

【解決手段】 主処理にあたるプログラムコード部分が暗号化されているので、第三者はデバイスドライバを逆アセンブラで解析することはできない。初期化処理 4 1 において、この暗号化コード 4 2 は復号化される。このためデバイス使用時には、暗号化コード 4 2 は元のプログラムコードに戻っているため、通常の処理を行うことができる。デバイス解放時には終了処理 4 3 において、プログラムコードから暗号化コード 4 2 に戻される。暗号化の解除を主処理を実行する直前に行い、再暗号化を主処理を実行した直後にしてもよい。これにより、実行速度が低下する代わりに、暗号化が解除されている時間が非常に短くなるため、解析されにくいドライバを構築することができる。

【選択図】 図 4